**Project overview/summary:**

**Technology summary:**

|  |  |  |
| --- | --- | --- |
| **Streaming** | **Kafka** | **AWS Cluster (1 zookeeper , 1 Broker)** |
| **Producer** | **Python Kafka** | **Deployed to Broker** |
| **Big data platform** | **Spark 2.1** | **AWS EMR (1 Master , 2 Core servers)** |
| **Consumer** | **Pyspark kafka** | **AWS EMR step executed client mode** |
| **Data storage** | **Cassandra 3.9, cqlsh 5.0.1** | **AWS Cassandra server** |
| **Data Analysis** |  |  |
| **Local desktop** | **Windows 10** |  |

**Infrastructure set up overview:**

* Cloud – AWS free account
* Kafka cluster

AWS Cloud formation used to create Kafka cluster with 1 zookeeper server and 1 Broker server; Cloud formation has a template to provision these servers pre-configured with all settings required for kafka streaming like adding zookeeper server in server.properties of kafka broker

SSH into zookeeper server to create 1 topic “airline1” for this project

More details on set up below

* Spark cluster

* Cassandra cluster

**Design-Code overview:**

**Execution demo:**

**Infrastructure set up details(Kafka):**

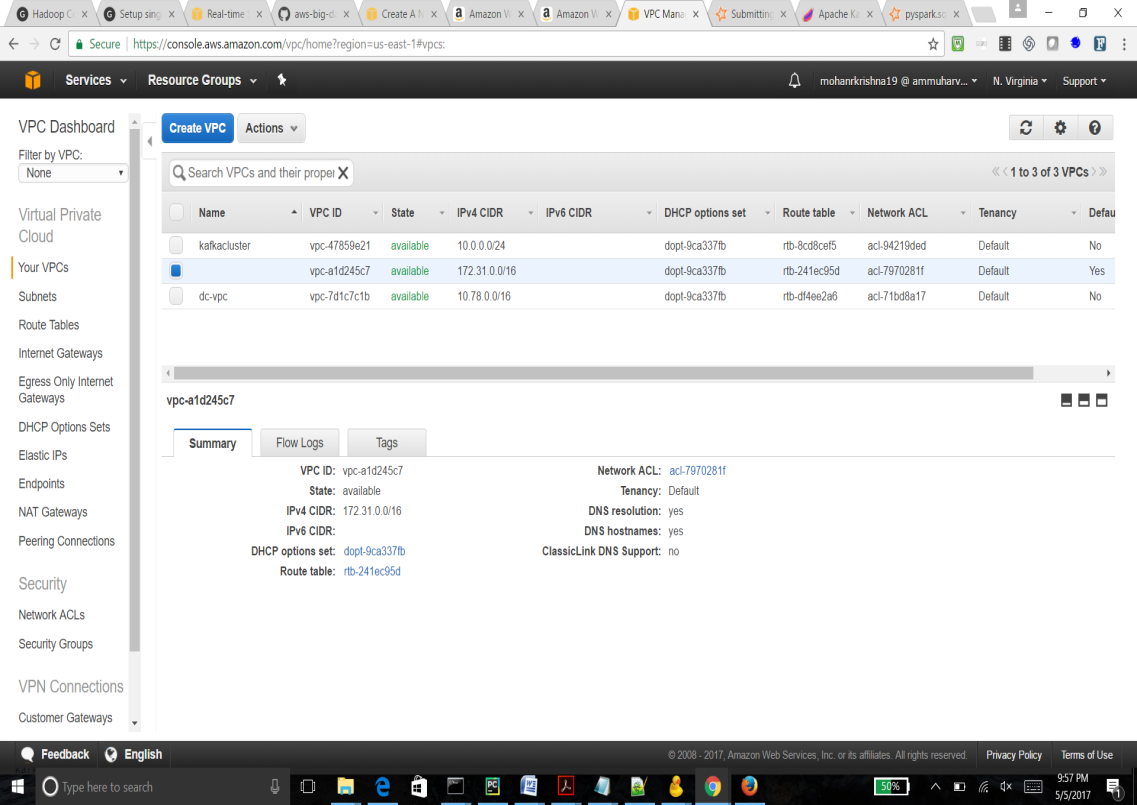
**Kafka cluster:**

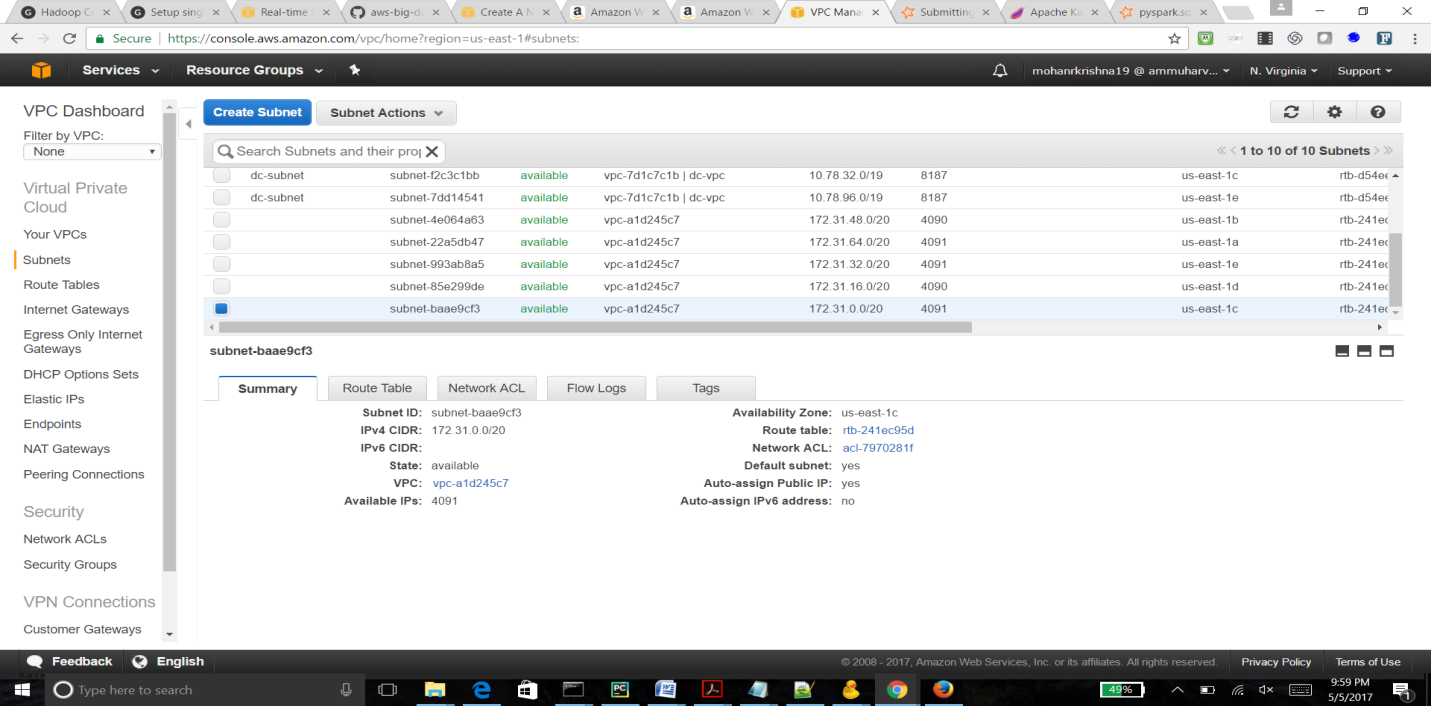
* Create following from AWS VPC to create Virtual private cloud and subnet for creating Kafka cluster
  + Create EC2 keyvalue pair for ssh; Download .pem file to local desktop
  + Create VPC and Subnet ; Important settings –

(VPC) DNS resolution: yes ; DNS hostnames:yes

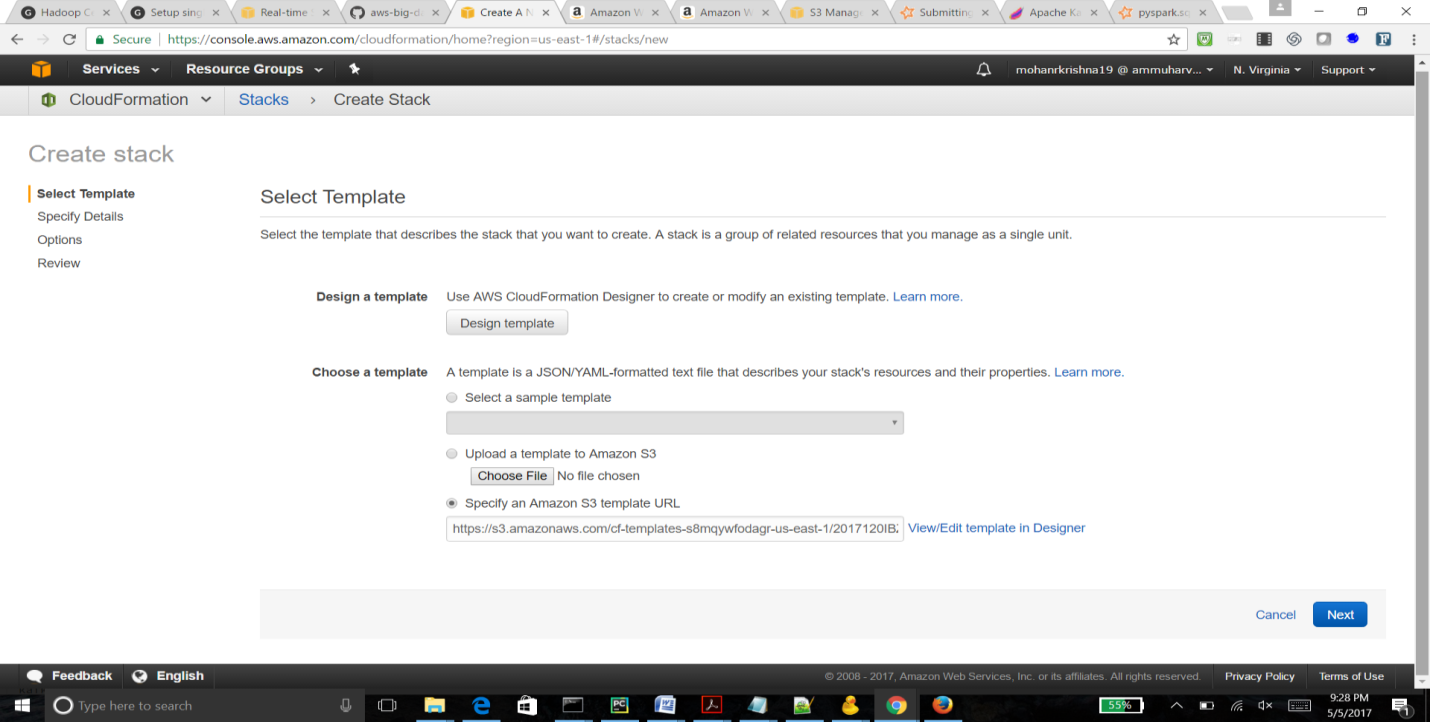
(Subnet) Auto-assign Public IP: Yes

|  |  |
| --- | --- |
|  |  |

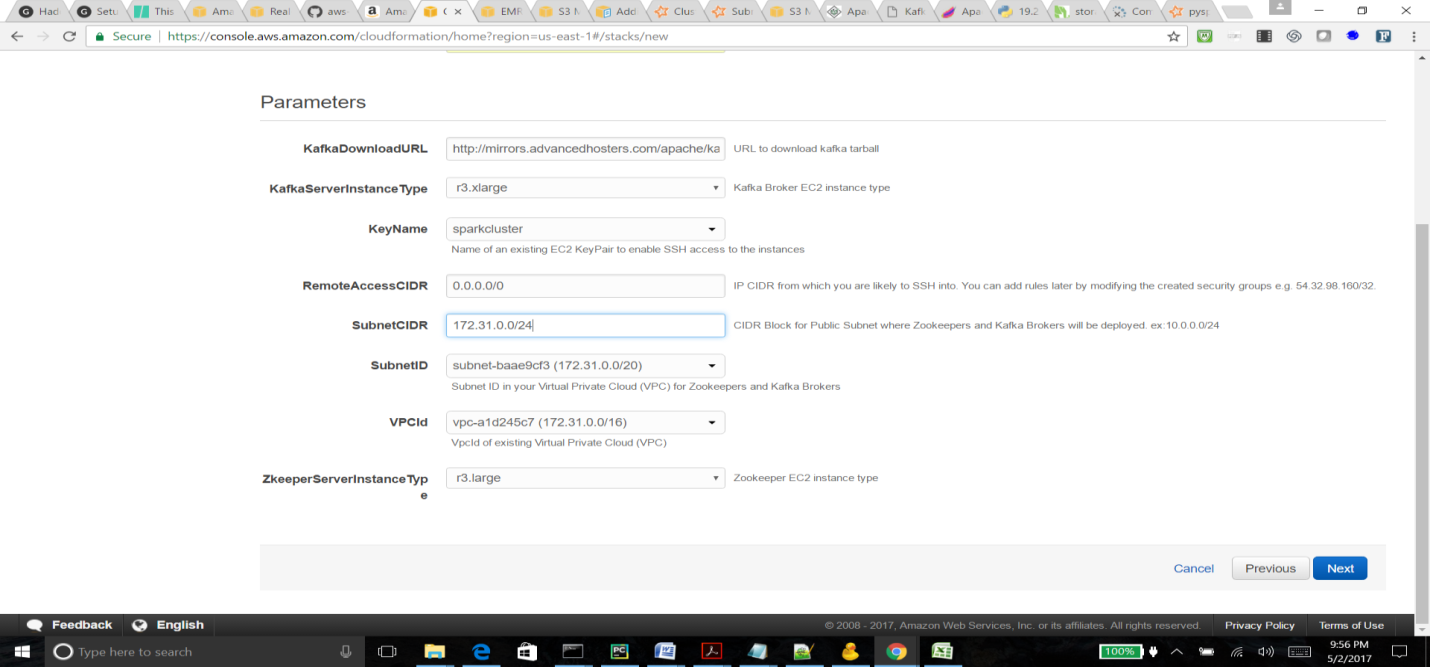




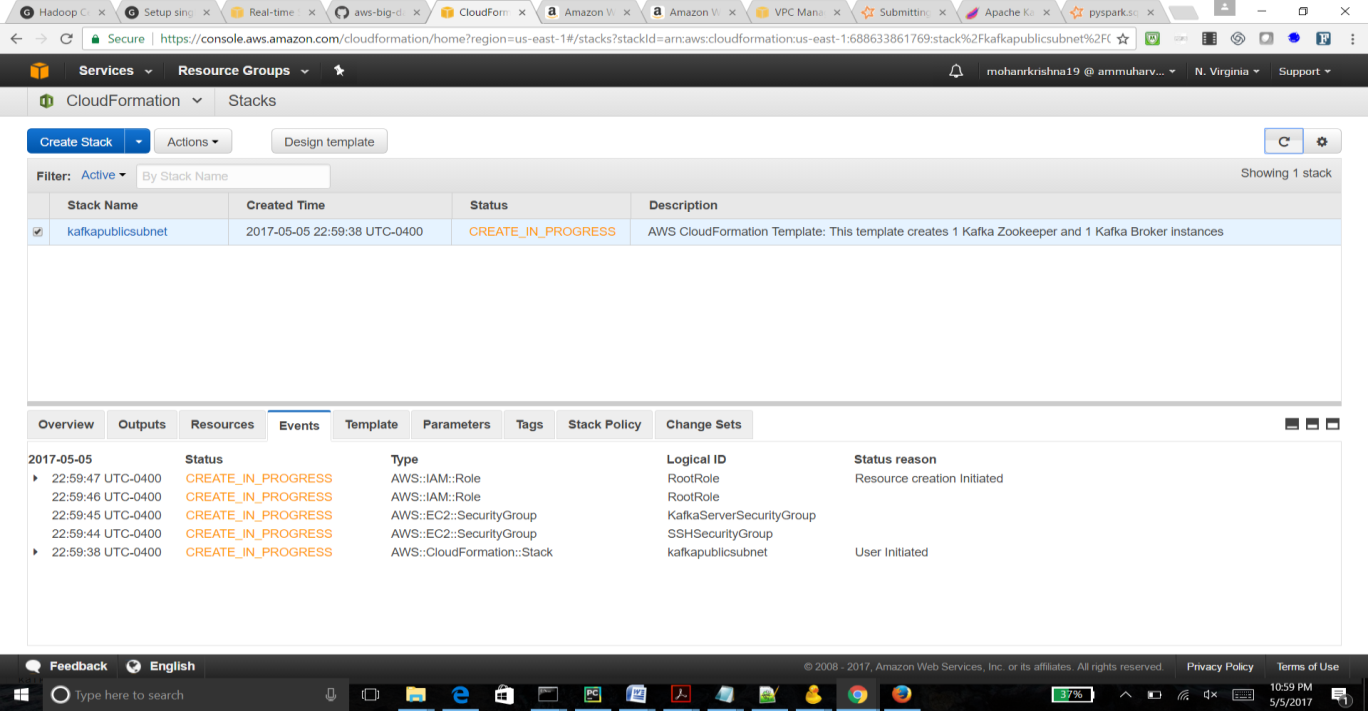
* Use json template at [CloudFormation template for public subnets](https://github.com/awslabs/aws-big-data-blog/blob/master/aws-blog-sparkstreaming-from-kafka/cloudformation/SetupKafka-AWS-PublicSubnets.json) ; Download to s3
* From AWS Cloud formation , create stack , choose above s3 json



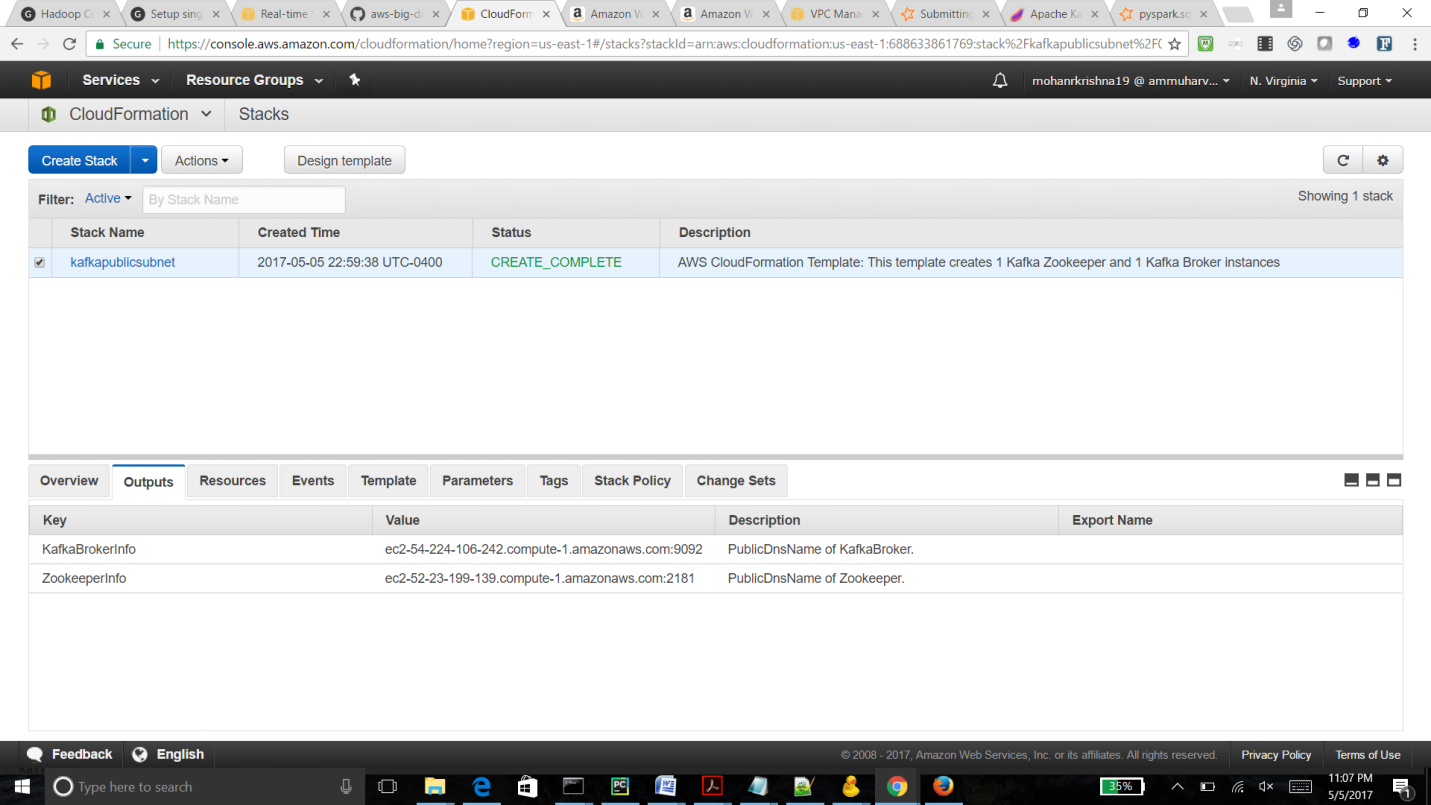
* Give “stackname” kafkapublicsubnet ; Select VPCid and subnet created above; Choose KeyName=EC2 keyvalue pair created. Important : SubnetCIDR should match Subnet CIDR , otherwise servers cannot be connected ; Keep Remote CIDR to 0.0.0.0/0 ; this is for laptops from which SSH is invoked. SourceIP can be added later too from EC2 menu



* Below screen shows progress , wait until ‘CREATE\_COMPLETE’



* Navigate to Outputs and save publicdns of KafkaBroker , Zookeeper servers , Required to login using SSH



**Kafka cluster create topic:**

* Install Git bash desktop app on windows
* Using Git bash , cd to directory with EC2 private key and remote login using SSH

email@LAPTOP-9CHTJ9IO MINGW64 /D/BigdataVMWare/Aws Keys

**$ ssh -i "sparkcluster.pem" ec2-user@ec2-52-23-199-139.compute-1.amazonaws.com**

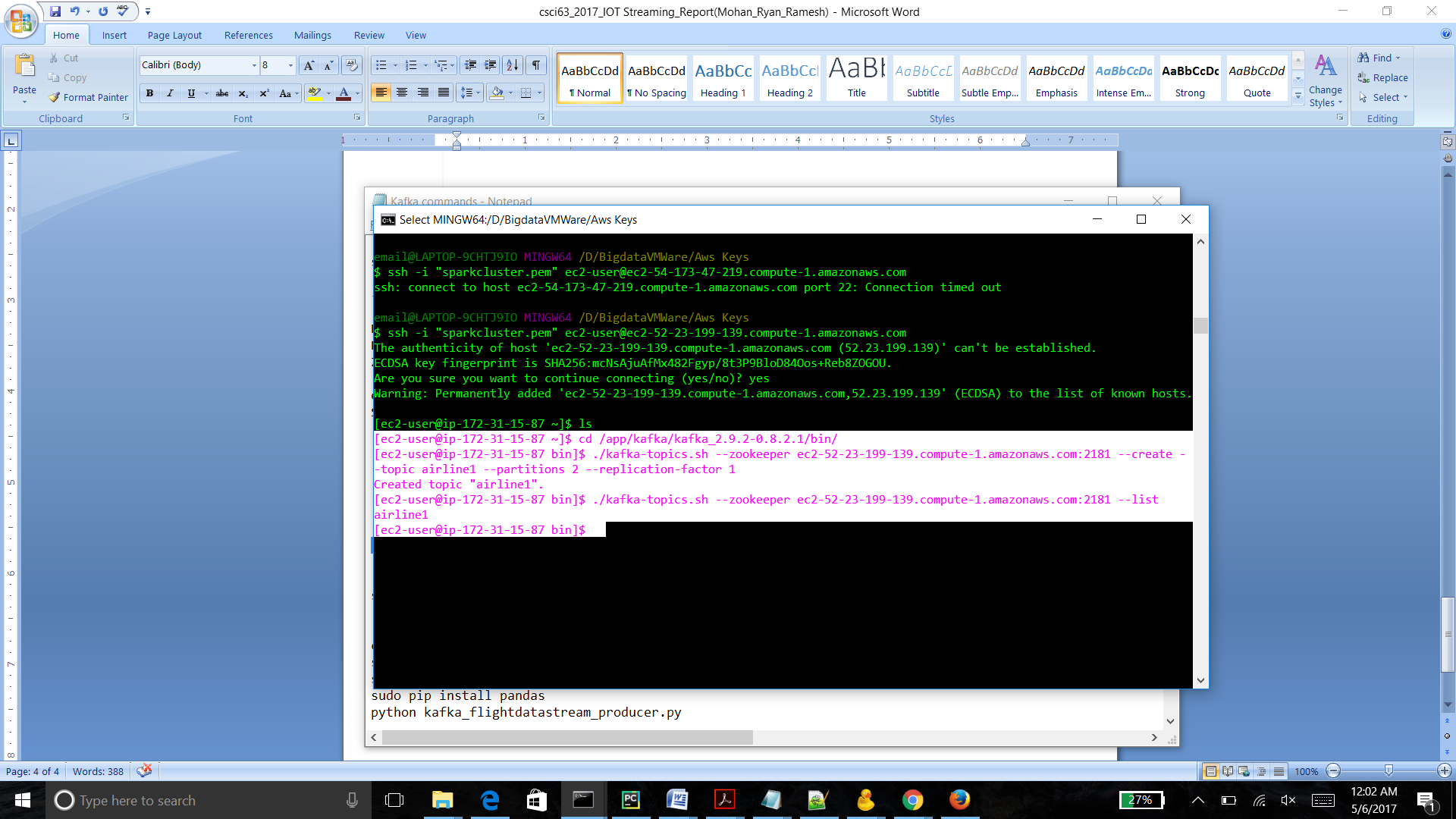
The authenticity of host 'ec2-52-23-199-139.compute-1.amazonaws.com (52.23.199.139)' can't be established.

ECDSA key fingerprint is SHA256:mcNsAjuAfMx482Fgyp/8t3P9BloD84Oos+Reb8ZOGOU.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'ec2-52-23-199-139.compute-1.amazonaws.com,52.23.199.139' (ECDSA) to the list of known hosts

* Navigate to kafka bin directory and create topic ; Verify topic created using --list

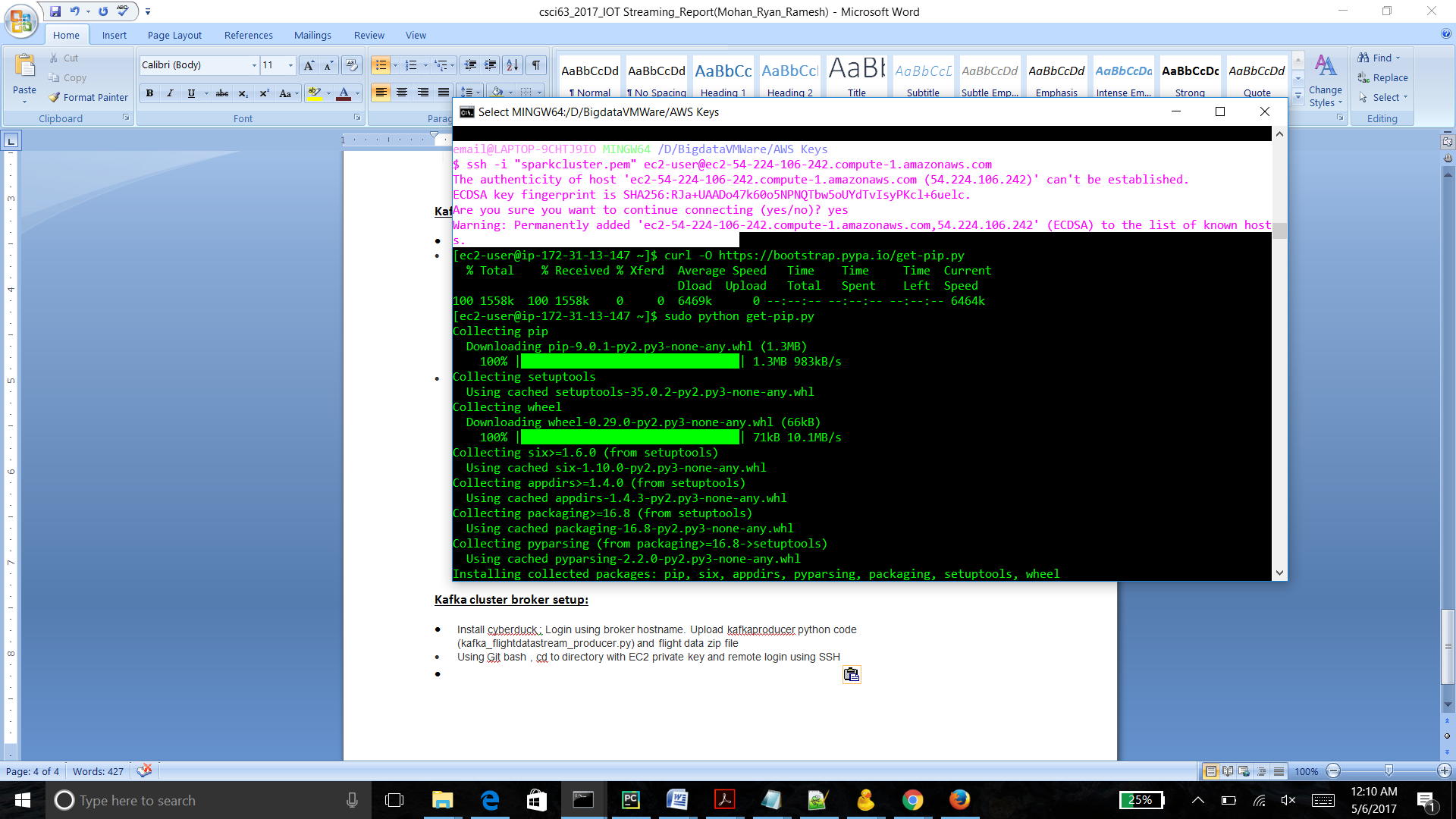


**Kafka cluster broker setup:**

* Install cyberduck ; Login using broker hostname. Upload kafkaproducer python code (kafka\_flightdatastream\_producer.py) and flight data zip file
* Using Git bash , cd to directory with EC2 private key and remote login using SSH

**email@LAPTOP-9CHTJ9IO MINGW64 /D/BigdataVMWare/AWS Keys**

**$ ssh -i "sparkcluster.pem"** [**ec2-user@ec2-54-224-106-242.compute-1.amazonaws.com**](mailto:ec2-user@ec2-54-224-106-242.compute-1.amazonaws.com)



* Kafka broker is used to run producer python program. So below python packages are required. Pip needs to be installed first followed by kafka-python , pandas

curl -O https://bootstrap.pypa.io/get-pip.py

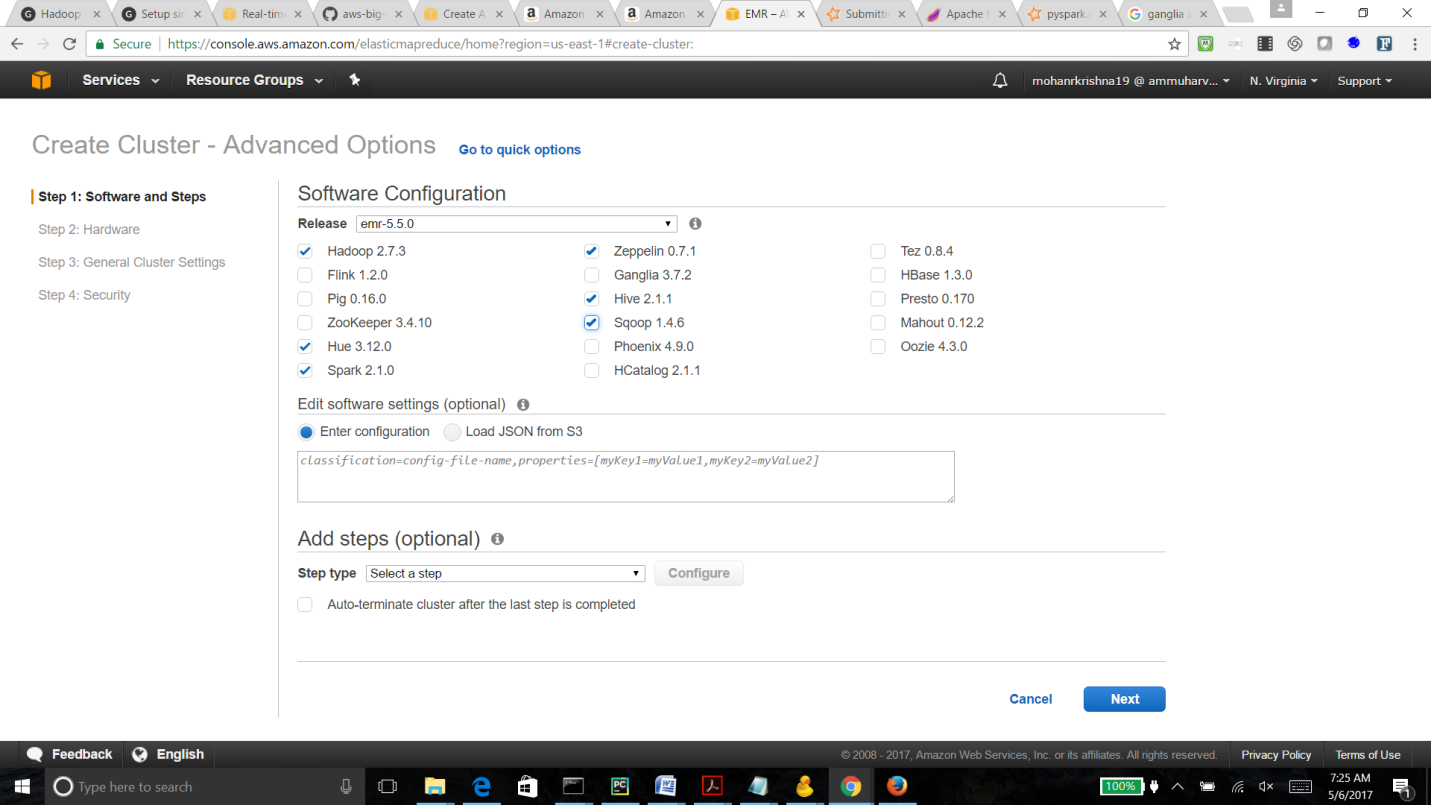
sudo python get-pip.py

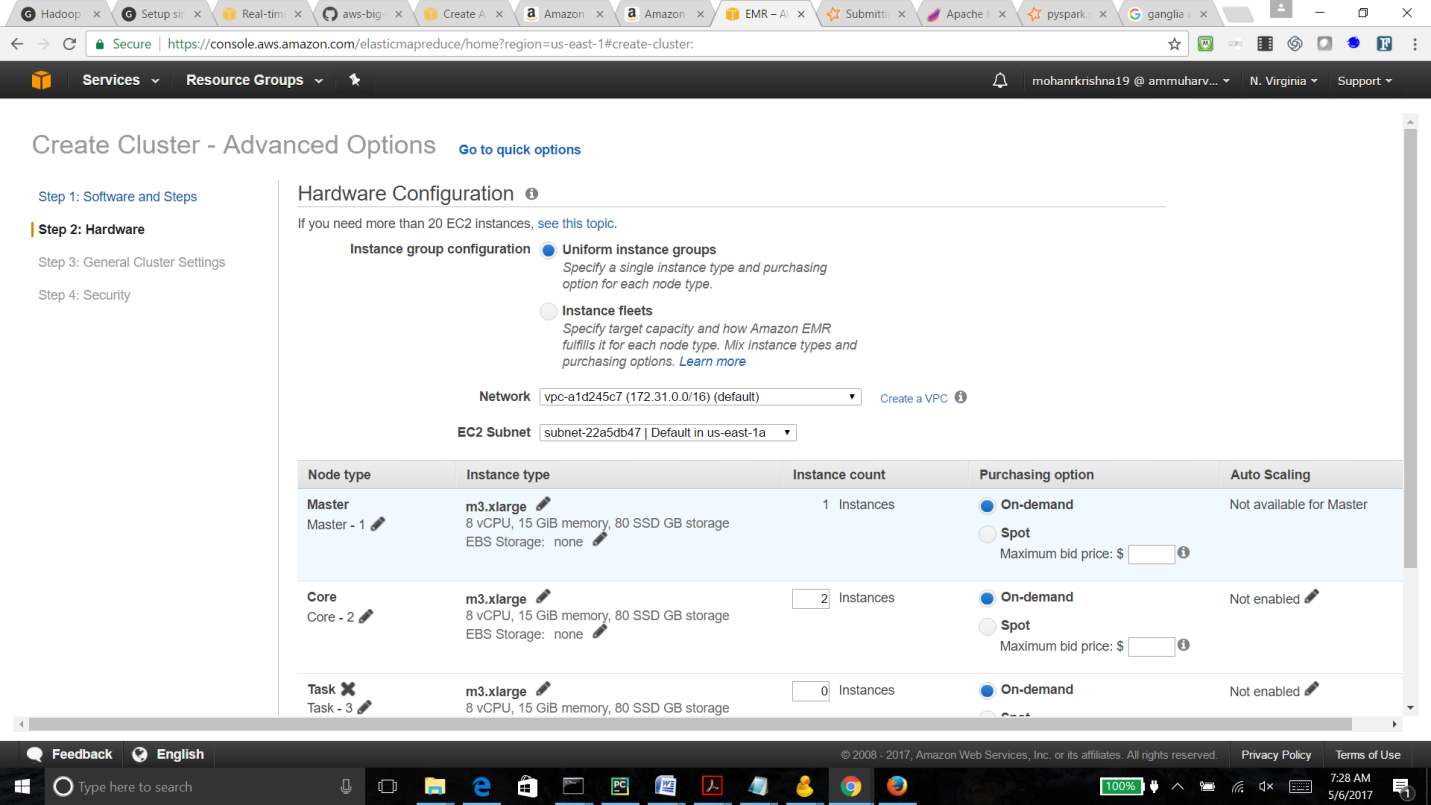
sudo pip install kafka-python

sudo pip install pandas

**Infrastructure set up details(Spark):**

* Navigate to Amazon EMR (AWS)
* Click Create cluster ; give name of cluster – ‘sparkcluster’ and go to advanced options
* Choose latest emr – emr-5.5.0 , hadoop is default selected and required for spark to use HDFS , Yarn ; Choose Spark 2.1.0



* Next hardware configuration , as this is small scale and pilot choose low scale servers to avoid charges ; Master node – m3.xlarge(8core , 15GB) , 2 Cores (for Spark Worker nodes, 2 Core , 3.8GB) ; Choose Network VPC and Subnet same as Kafka cluster to be able to avoid cross network access issues between spark and kafka servers
* Next general options , leave as-is
* Next Security Options , choose EC2 key value pair. I chose the same one used for Kafkacluster; Suggested to use different one
* Click Create cluster; It would take 15-20 mins